Approximating Bin Packing within $O(\log OPT \cdot \log\log OPT)$ bins

Thomas Rothvoß*

Massachusetts Institute of Technology

Abstract

For *bin packing*, the input consists of n items with sizes $s_1, \ldots, s_n \in [0,1]$ which have to be assigned to a minimum number of bins of size 1. The seminal Karmarkar-Karp algorithm from '82 produces a solution with at most $OPT + O(\log^2 OPT)$ bins.

We provide the first improvement in now 3 decades and show that one can find a solution of cost $OPT + O(\log OPT \cdot \log\log OPT)$ in polynomial time. This is achieved by rounding a fractional solution to the Gilmore-Gomory LP relaxation using the *Entropy Method* from discrepancy theory. The result is constructive via algorithms of Bansal and Lovett-Meka.

1 Introduction

Bin Packing is one of the very classical combinatorial optimization problems studied in computer science and operations research. It's study dates back at least to the 1950's [Eis57] and it appeared as one of the prototypical **NP**-hard problems in the book of Garey and Johnson [GJ79]. For a detailed account, we refer to the survey of [CGJ84]. Bin Packing is also a good case study to demonstrate the development of techniques in approximation algorithms. The earliest ones are simple greedy algorithms such as the *First Fit* algorithm, analyzed by Johnson [Joh73] which requires at most $1.7 \cdot OPT + 1$ bins and *First Fit Decreasing* [JDU+74], which yields a solution with at most $\frac{11}{9} OPT + 4$ bins (see [Dós07] for a tight bound of $\frac{11}{9} OPT + \frac{6}{9}$). Later, Fernandez de la Vega and Luecker [FdlVL81] developed an *asymptotic PTAS* by introducing an *item grouping technique* that reduces the number of different item types and has been reused in numerous papers for related problems. De la Vega and Luecker were able to find a solution of cost at most $(1+\varepsilon)OPT + O(\frac{1}{\varepsilon^2})$ for Bin Packing and the running time is either of the form $O(n^{f(\varepsilon)})$ if one uses dynamic programming or of the form $O(n \cdot f(\varepsilon))$ if one applies linear programming techniques.

A big leap forward in approximating bin packing was done by Karmarkar and Karp in 1982 [KK82], who provided an iterative rounding approach for the mentioned linear programming formulation which produces a solution with at most $OPT + O(\log^2 OPT)$ bins in polynomial time, corresponding to an *asymptotic FPTAS*.

*Email: rothvoss@math.mit.edu. Supported by the Alexander von Humboldt Foundation within the Feodor Lynen program, by ONR grant N00014-11-1-0053 and by NSF contract CCF-0829878.

Both papers [FdlVL81, KK82] used the *Gilmore-Gomory LP relaxation* (see e.g. [Eis57, GG61])

$$\min\left\{\mathbf{1}^T x \mid Ax = \mathbf{1}, x \geq \mathbf{0}\right\} \tag{1}$$

where A is the *pattern matrix* that consists of all column vectors $\{p \in \mathbb{Z}^n_{\geq 0} \mid p^T s \leq 1\}$. Each such column p is called a *(valid) pattern* and corresponds to a feasible multiset of items that can be assigned to a single bin. Note that it would be perfectly possible to consider a stronger variant in which only patterns $p \in \{0,1\}^n$ are admitted. In this case, the LP (1) could also be interpreted as the standard (UNWEIGHTED) SET COVER relaxation

$$\min\left\{\sum_{S\in\mathscr{S}} x_S \mid \sum_{S\in\mathscr{S}:i\in S} x_S \geq 1\ \forall i \in [n]; x_S \geq 0\ \forall S \in \mathscr{S}\right\} \tag{2}$$

for the set system $\mathscr{S} := \{S \subseteq [n] \mid \sum_{i\in S} s_i \leq 1\}$. However, the additive gap between both versions is at most $O(\log n)$ anyway, thus we stick to the matrix-based formulation as this is more suitable for our technique[1].

Let OPT and OPT_f be the value of the best integer and fractional solution for (1) respectively. Although (1) has an exponential number of variables, one can compute a basic solution x with $\mathbf{1}^T x \leq OPT_f + \delta$ in time polynomial in n and $1/\delta$ [KK82] using the Grötschel-Lovász-Schrijver variant of the Ellipsoid method [GLS81]. Alternatively, one can also use the Plotkin-Shmoys-Tardos framework [PST95] or the multiplicative weight update method (see e.g. the survey of [AHK12]) to achieve the same guarantee.

The Karmarkar-Karp algorithm operates in $\log n$ iterations in which one first groups the items such that only $\frac{1}{2}\sum_{i\in[n]} s_i$ many different item sizes remain; then one computes a basic solution x and buys $\lfloor x_p \rfloor$ times pattern p and continues with the residual instance. The analysis provides a $O(\log^2 OPT)$ upper bound on the *additive* integrality gap of (1). In fact, it is even conjectured in [ST97] that (1) has the *Modified Integer Roundup Property*, i.e. $OPT \leq \lceil OPT_f \rceil + 1$ (and up to date, there is no known counterexample; the conjecture is known to be true for instances that contain at most 7 different item sizes [SS09]). Recently, [EPR11] found a connection between coloring permutations and bin packing which shows that *Beck's Three Permutations Conjecture* (any 3 permutations can be bi-colored with constant discrepancy) would imply a constant integrality gap at least for instances with all item sizes bigger than $\frac{1}{4}$. Note that the gap bound of the Karmarkar-Karp algorithm is actually of the form $O(\log OPT_f \cdot \log(\max_{i,j}\{\frac{s_i}{s_j}\}))$, which is $O(\log n)$ for such instances. But very recently Newman and Nikolov [NNN12] found a counterexample to Beck's conjecture.

Considering the gap that still remains between upper and lower bound on the additive integrality gap, one might be tempted to try to modify the Karmarkar-Karp algorithm in order to improve the approximation guarantee. From an abstract point of view, [KK82] buy only patterns that already appear in the initial basic solution x and then map every item to the slot of a *single* larger item. Unfortunately, combining the insights from [NNN12] and [EPR11], one can show that no algorithm with this abstract property can yield a $o(\log^2 n)$ gap, which establishes a barrier for a fairly large class of algorithms [EPR13].

A simple operation that does not fall into this class is the following:

[1] For example, if the input consists of a single item of size $\frac{1}{k}$, then the optimum value of (2) is 1, while the optimum value of (1) is $\frac{1}{k}$. But the additive gap can be upper bounded as follows: Take a solution x to (1) and apply a single grouping via Lemma 6 with parameter $\beta = 1$. This costs $O(\log n)$ and results in a solution to (2) for some general right hand side vector $b \in \mathbb{Z}^n_{\geq 0}$. With the usual cloning argument, this can be easily converted into the form with right hand side $\mathbf{1}$.

Gluing: Whenever we have a pattern p with $x_p > 0$ that has many copies of the same item, glue these items together and consider them as a single item.

In fact, iterating between gluing and grouping, results in a mapping of *several* small input items into the slot of a *single* large item – the barrier of [EPR13] does not hold for such a rounding procedure.

But the huge problem is: there is no reason why in the worst case, a fractional bin packing solution x should contain patterns with many items of the same type. Also the Karmarkar-Karp rounding procedure does not seem to benefit from that case either. However, there is an alternative algorithm of the author [Rot12] to achieve a $O(\log^2 OPT)$ upper bound, which is based on *Beck's entropy method* [Bec81, Bec88, Spe85] (or *partial coloring lemma*) from *discrepancy theory*. This is a subfield of combinatorics which deals with the following type of questions: given a set system $S_1, \ldots, S_n \subseteq [m]$, find a coloring of the elements $1, \ldots, m$ with red and blue, such that for each set S_i the difference between the red and blue elements (called the *discrepancy*) is as small as possible.

2 Outline of the technique

The partial coloring method is a very flexible technique to color at least half of the elements in a set system with a small discrepancy, but the technique is based on the pigeonhole principle — with exponentially many pigeons and pigeonholes — and is hence non-constructive in nature[2]. But recently Bansal [Ban10] and later Lovett and Meka [LM12] provided polynomial time algorithms to find those colorings. In fact, it turns out that our proofs are even simpler using the Lovett-Meka algorithm than using the classical non-constructive version, thus we directly use the constructive method.

The constructive partial coloring lemma

The Lovett-Meka algorithm provides the following guarantee[3]:

Lemma 1 (Constructive partial coloring lemma [LM12])**.** *Let $x \in [0,1]^m$ be a starting point, $\delta > 0$ an arbitrary error parameter, $v_1, \ldots, v_n \in \mathbb{Q}^m$ vectors and $\lambda_1, \ldots, \lambda_n \geq 0$ parameters with*

$$\sum_{i=1}^{n} e^{-\lambda_i^2/16} \leq \frac{m}{16}. \tag{3}$$

Then there is a randomized algorithm with expected running time $O(\frac{(m+n)^3}{\delta^2} \log(\frac{nm}{\delta}))$ to compute a vector $y \in [0,1]^m$ with

[2] The claim of the Partial coloring lemma is as follows: Given any vectors $v_1, \ldots, v_n \in \mathbb{R}^m$ with a parameters $\lambda_1, \ldots, \lambda_n > 0$ satisfying

$$\sum_{i=1}^{n} G(\lambda_i) \leq \frac{m}{5}, \quad \text{for} \quad G(\lambda) := \begin{cases} 9e^{-\lambda^2/5} & \text{if } \lambda \geq 2 \\ \log_2(32 + \frac{64}{\lambda}) & \text{if } \lambda < 2 \end{cases}$$

Then there is a partial coloring $\chi : [m] \to \{0, \pm 1\}$ with $|\mathrm{supp}(\chi)| \geq \frac{m}{2}$ and $|v_i \chi| \leq \lambda_i \|v_i\|_2$ for all vectors $i = 1, \ldots, n$.

[3] The original statement has $x, y \in [-1,1]^m$ and $|y_j| \geq 1 - \delta$ for half of the entries. However, one can obtain our version as follows: Start with $x \in [0,1]^m$. Then apply [LM12] to $x' := 2x - \mathbf{1} \in [-1,1]^m$ to obtain $y' \in [-1,1]^m$ with $|v_i(y' - x')| \leq \lambda_i \|v_i\|_2$ for all $i \in [n]$ and half of the entries satisfying $|y'_j| \geq 1 - \delta$. Then $y := \frac{1}{2}(y' + \mathbf{1})$ has half of the entries $y_j \in [0, \frac{\delta}{2}] \cup [1 - \frac{\delta}{2}, 1]$. Furthermore, $|v_i(y - x)| = \frac{1}{2}|v_i(y' - x')| \leq \frac{\lambda_i}{2} \|v_i\|_2$.

- $y_j \in [0,\delta] \cup [1-\delta, 1]$ *for at least half of the indices* $j \in \{1,\dots,m\}$
- $|v_i y - v_i x| \leq \lambda_i \cdot \|v_i\|_2$ *for each* $i \in \{1,\dots,n\}$.

If we end up with an almost integral Bin Packing solution y, we can remove all entries with $y_j \leq \frac{1}{n}$ and roundup those with $y_j \in [1-\frac{1}{n}, 1]$ paying only an additional constant term. Thus we feel free to ignore the δ term and assume that half of the entries are $y_j \in \{0,1\}$.

The algorithm in [LM12] is based on a simulated Brownian motion in the hypercube $[0,1]^m$ starting at x. Whenever the Brownian motion hits either the boundary planes $y_j = 0$ or $y_j = 1$ or one of the hyperplanes $v_i(x-y) = \pm\lambda_i \|v_i\|_2$, the Brownian motion continues the walk in that subspace. By standard concentration bounds, the probability that the walk ever hits the ith hyperplane is upperbounded by $e^{-\Omega(\lambda_i^2)}$. In other words, condition (3) says that the expected number of hyperplanes $v_i(x-y) = \pm\lambda_i \|v_i\|_2$ that ever get hit is bounded by $\frac{m}{16}$, from which one can argue that a linear number of boundary constraints must get tight.

Readers that are more familiar with approximation algorithm techniques than with discrepancy theory, should observe the following: In the special case that $\lambda_i = 0$ for all i, one can easily prove Lemma 1 by choosing y as any basic solution of $\{y \mid v_i y = v_i x \ \forall i \in [n];\ \mathbf{0} \leq y \leq \mathbf{1}\}$. In other words, Lemma 1 is somewhat an extension of the concept of basic solutions. Considering that a significant fraction of approximation algorithms is based on the sparse support of basic solutions, one should expect many more applications of [LM12].

The rounding procedure

Let x be a fractional solution for the Gilmore-Gomory LP (1), say with $|\text{supp}(x)| = m \leq n$ and let A be the constraint matrix reduced to patterns in the support of x.

Assume for the sake of simplicity that all items have size between $\frac{1}{k}$ and $\frac{2}{k}$ for some k. We now want to discuss how Lemma 1 can be applied in order to replace x with another vector y that has half of the entries integral and is still almost feasible. Then repeating this procedure for $\log(m)$ iterations will lead to a completely integral solution. For the sake of comparison: the Karmarkar-Karp algorithm is able to find another fractional y that has at most half the support of x and is at most an additive $O(1)$ term more costly. So let us argue how to do better.

Let us sort the items according to their sizes (i.e. $\frac{2}{k} \geq s_1 \geq \ldots \geq s_n \geq \frac{1}{k}$) and partition the items into *groups* $I_1,\dots,I_t$ such that the number of incidences in A is of order $100k$ for each group. In other words, if we abbreviate $v_{I_j} := \sum_{i \in I_j} A_i$ as the sum of the row vectors in I_j, then $\|v_{I_j}\|_1 \approx 100k$. Since each column of A sums up to at most k and each group consums $100k$ incidences, we have only $t \leq \frac{m}{100}$ many groups. Now, we can obtain a suitable y with at most half the fractional entries by either computing a basic solution to the system

$$v_{I_j}(x-y) = 0 \ \ \forall j \in [t], \quad \mathbf{1}^T y = \mathbf{1}^T x, \quad \mathbf{0} \leq y \leq \mathbf{1}$$

or by applying the Constructive Partial Coloring Lemma to $v_{I_1},\dots,v_{I_t}$ and $v_{\text{obj}} := (1,\dots,1)$ with a uniform parameter of $\lambda := 0$. In fact, since $(t+1)\cdot e^{-0^2/16} \leq \frac{m}{100} + 1 \leq \frac{m}{16}$, condition (3) is even satisfied with a generous slack. The meaning of the constraint $v_I(x-y) = 0$ is that y still contains the right number of slots for items in group I. But the constraint does not distinguish between different items within I; so maybe y covers the smaller items in I more often than needed and leaves the larger ones uncovered. However, it is not hard to argue that

after discarding $100k$ items, y can be turned into a feasible solution, so the increase in the objective function is again $O(1)$ as for Karmarkar-Karp.

Now we are going to refine our arguments and use the power of the entropy method. The intuition is that we want to impose stronger conditions on the coverage of items *within* groups. Consider a group $I := I_j$ and create growing *subgroups* $G_1 \subseteq G_2 \subseteq \ldots \subseteq G_{1/\varepsilon} = I$ such that the number of incidences grows by $\varepsilon \cdot 100k$ from subgroup to subgroup, for some $\varepsilon > 0$ (later $\varepsilon := \frac{1}{\log^2 n}$ will turn out to be a good choice; see Figure 1.(a)). In other words, $\|v_{G_{j+1}\setminus G_j}\|_1 \approx \varepsilon \cdot 100k$. We augment the input for Lemma 1 by the vectors v_G for all subgroups equipped with parameter $\lambda_G := 4\sqrt{\ln(\frac{1}{\varepsilon})}$. Observe that condition (3) is still satisfied as each of the $\frac{t}{\varepsilon}$ many subgroups G contributes only $e^{-\lambda_G^2/16} \leq \varepsilon$. So, we can get a better vector y that also satisfies $|v_G(x-y)| \leq 4\sqrt{\ln(\frac{1}{\varepsilon})} \cdot \|v_G\|_2$ for any subgroup. In order to improve over our previous approach we need to argue that $\|v_G\|_2 \ll k$. But we remember that by definition $\|v_G\|_1 \leq 100k$, thus we obtain $\|v_G\|_2 \leq \sqrt{\|v_G\|_1 \cdot \|v_G\|_\infty} \leq 100k \cdot \sqrt{\|v_G\|_\infty / \|v_G\|_1}$. In other words, the only situation in which we do not immediately improve over Karmarkar-Karp is if $\|v_G\|_\infty \geq \Omega(\|v_G\|_1)$, i.e. if there is some pattern such that a large fraction of it is filled with items of the same subgroup G.

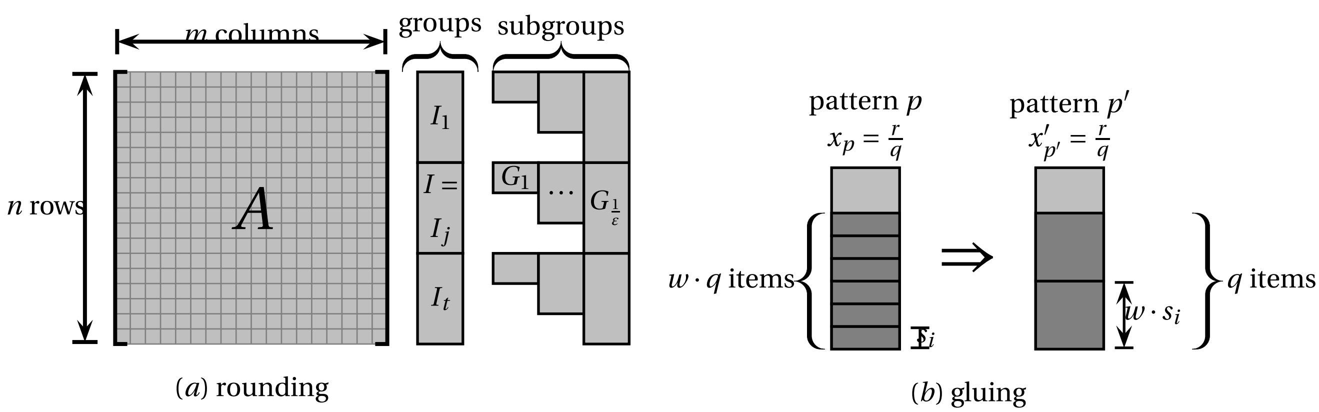


Figure 1: (a) visualization of groups and subgroups. (b) visualization of the gluing procedure.

The gluing

At this point our gluing operation comes into play. After a simple *pre-rounding step* which costs us a $o(1)$ term, we can assume that all entries in x are multiples of $\frac{1}{q} := \frac{1}{\log^4 n}$. Recall that initially we have a single copy from each item. We group consecutive items together into groups of size $\beta = \frac{1}{\log^4 n}$ and round their sizes to the smallest one in the group. By standard arguments this incurs a negligible cost of $O(\frac{1}{\log^4 n})$. Now we can assume that we have a sufficiently large number of copies for every item. Suppose that after this agglomeration we find an item i and a pattern p in the support such that indeed p_i is large, say $p_i s_i \geq \frac{1}{\log^8 n}$. The crucial observation is that this pattern p alone covers $w := \lfloor \frac{p_i}{q} \rfloor$ many copies of item i in the input since $x_p \cdot p_i \geq w$. Next, take w many copies of item i in p and glue them together to obtain a new, bigger item i' of size $s_{i'} = w \cdot s_i$. The pattern p has enough items to do this

q times, see Figure 1.(b). In other words, the modified pattern now contains q copies of a new artificial item i'. The reason why we want q copies of this new item is that the modified pattern p alone covers $q \cdot x_p = 1$ copies of i'. Thus, in a finally obtained integral solution we would have a slot for the artificial item i', which we can then replace with the copies of the original item i.

Observe that the size of this newly obtained item type is $s_{i'} = w \cdot s_i \geq \frac{1}{\log^{12} n}$. So we call items above that size *large* and below that size *small*. The interesting effect is that if we apply this gluing procedure to all small items whenever possible, the penalty that we pay for rounding the remaining small items is so small that the overall cost is completely dominated by the contribution of the large items (i.e. those items that were either large from the beginning or that were created during the gluing process). In other words, we obtain the same approximation guarantee as if the instance would only contain items of size at least $\frac{1}{\log^{12} n}$ from the beginning on; for those instances already [KK82] produces a solution with at most $OPT_f + O(\log n \cdot \log\log n)$ bins, so this is our final approximation guarantee for all instances.

Contribution

Our main contribution is the following theorem:

Theorem 2. *For any Bin Packing instance $s_1, \ldots, s_n \in [0,1]$, one can compute a solution with at most $OPT_f + O(\log OPT_f \cdot \log\log OPT_f)$ bins in expected time $O(n^6 \log^5(n))$, where OPT_f denotes the optimum value of the Gilmore-Gomory LP relaxation.*

This partly solves problem #3 in the list of 10 open problems in approximation algorithms stated by Williamson and Shmoys [WS11] (they asked for a constant integrality gap).

3 Related work

The classical application of the partial coloring lemma is to find a coloring $\chi : [m] \to \{\pm 1\}$ for m elements such that for a given set system $S_1, \ldots, S_n$[4] the *discrepancy* $\max_{i \in [n]} |\sum_{j \in S_i} \chi(j)|$ is minimized. For example, one can obtain Spencer's bound [Spe85] on the discrepancy of arbitrary set systems, by applying $\log m$ times Lemma 1 starting with $x := (\frac{1}{2}, \ldots, \frac{1}{2})$ and a uniform bound of $\lambda := C\sqrt{\log \frac{2n}{m}}$ where $v_i \in \{0,1\}^m$ is the characteristic vector of S_i. This results in a coloring $\chi : [m] \to \{\pm 1\}$ with $|\chi(S)| \leq O(\sqrt{m \log \frac{2n}{m}})$. Note that e.g. for $n \leq O(m)$, this is a $O(\sqrt{m})$ coloring, while a pure random coloring would be no better than $O(\sqrt{m \cdot \log m})$.

Other applications of this method give a $O(\sqrt{t} \log m)$ bound if no element is in more than t sets [Sri97] and a $O(\sqrt{k} \log m)$ bound for the discrepancy of k permutations [SST]. For the first quantity, alternative proof techniques give bounds of $2t-1$ [BF81] and $O(\sqrt{t \cdot \log m})$ [Ban98].

In fact, we could use those classical techniques and extend [Rot12] to obtain a $OPT_f + O(\log OPT_f \cdot \log\log OPT_f)$ *integrality gap* result. It might appear surprising that one can bound integrality gaps by coloring matrices, but this is actually a well known fact, which is expressed by the Lovász-Spencer-Vesztergombi Theorem [LSV86]: Given a matrix A and a

[4]The standard notation in discrepancy theory is to have n as number of elements and m as the number of sets. However, that conflicts with the standard notation for Bin Packing, where n is the number of items which is essentially the number of v-vectors.

vector $x \in [0,1]^m$ such that any submatrix of A admits a discrepancy α coloring. Then there is a $y \in \{0,1\}^m$ with $\|Ax - Ay\|_\infty \le \alpha$. For a more detailed account on discrepancy theory, we recommend Chapter 4 in the book of Matoušek [Mat99].

4 Preliminaries

In the Bin Packing literature, it is well known that it suffices to show bounds as in Theorem 2 with an n instead of OPT_f and that one can also assume that items are not too tiny, e.g. $s_i \ge \frac{1}{n}$. Though the following arguments are quite standard (see e.g. [KK82]), we present them for the sake of completeness.

Lemma 3. *Assume for a monotone function f, there is a poly(m)-time $OPT_f + f(n)$ algorithm for Bin Packing instances $s \in [0,1]^m$ with $|\{s_i \mid i \in [m]\}| \le n$ many different item sizes and $\min\{s_i \mid i \in [m]\} \ge \frac{1}{n}$. Then there is a polynomial time algorithm that finds a solution with at most $OPT_f + f(OPT_f) + O(\log OPT_f)$ bins.*

Proof. Let $s \in [0,1]^m$ be any bin packing instance and define $\sigma := \sum_{i=1}^m s_i$ as their size. First, split items into large ones $L := \{i \in [m] \mid s_i \ge \frac{1}{\sigma}\}$ and small ones $S := \{i \in [m] \mid s_i < \frac{1}{\sigma}\}$.

We perform the grouping procedure from [KK82] (or from Lemma 6) to large items L and produce an instance with sizes s' such that each size s'_i that appears has $\sum_{j: s'_j = s'_i} s'_i \ge 1$. Moreover, after discarding items of total size at most $O(\log \frac{1}{\min\{s_i \mid i \in L\}}) \le O(\log \sigma)$ one has $OPT'_f \le OPT_f$. Thus the number of different item sizes in s' is bounded by σ. We run the assumed algorithm to assign items in L to at most $OPT'_f + f(\sigma) \le OPT_f + f(OPT_f)$ bins (using that $OPT_f \ge \sigma$ and f is monotone). Adding the discarded items increases the objective function by at most another $O(\log OPT_f)$ term. Now we assign the small items greedily over those bins. If no new bin needs to be opened, we are done. Otherwise, we know that the solution consists of k bins such that $k-1$ bins are at least $1 - \frac{1}{\sigma}$ full. This implies $\sigma \ge (k-1)\cdot(1-\frac{1}{\sigma})$, and hence $k \le \sigma + 3 \le OPT_f + 3$ assuming $\sigma \ge 2$. $\square$

From now on, we have the implicit assumption $s_i \ge \frac{1}{n}$. In an alternative Bin Packing definition, also called the *cutting stock problem*, the input consists of a pair (s, b) such that $b_i \in \mathbb{Z}_{\ge 0}$ gives the number of copies of s_i. If b is unary encoded, this problem version is polynomially equivalent to classical Bin Packing. Otherwise, it is even unknown whether for $n = 3$ the problem admits a polynomial time algorithm (for $n = 2$ it can be solved in polynomial time and for any constant n, there is a polynomial time $OPT + 1$ algorithm [JSO10]). Moreover one cannot expect to even solve the LP (1) up to an *additive* constant in polynomial time. But the Karmarkar Karp bound of $O(\log^2 n)$ on the additive integrality gap still holds true (when n is the number of item types). In this paper, we will work with a more general formulation in which any $b \in \text{cone}\{p \in \mathbb{Z}^n_{\ge 0} \mid s^T p \le 1\}$ may serve as vector of multiplicities (note that such a vector might have fractional entries). From our starting solution x, we can immediately remove the integral parts $\lfloor x_p \rfloor$ and assume that $\mathbf{0} \le x < \mathbf{1}$, which has the consequence that $\sum_{i=1}^n s_i b_i < n$.

It will be useful to reformulate bin packing as follows: consider a size vector $s \in [0,1]^n$ $(s_1 \ge \ldots \ge s_n)$ with pattern matrix A and a given vector $x \in \mathbb{R}^m_{\ge 0}$ as input and aim to solve the

following problem

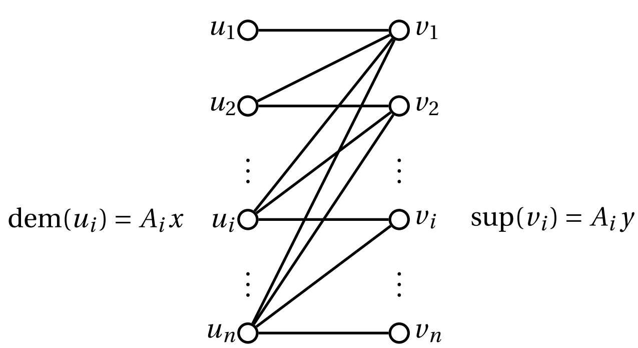
$$\begin{aligned} &\min \mathbf{1}^T y && \quad (4)\\ \sum_{j\le i} A_j y \;&\ge\; \sum_{j\le i} A_j x \quad \forall i\in[n] \\ y \;&\in\; \mathbb{Z}^m_{\ge 0} \end{aligned}$$

We write $y \succeq x$ if $\sum_{j\le i} A_j y \ge \sum_{j\le i} A_j x$ for all $i\in[n]$. In words: we have a fractional solution x to LP (1) for an instance with $A_i x$ many items of type i in the input and aim to find an integral solution y that reserves $\sum_{j\le i} A_j y$ many slots for items of type $1,\dots,i$. The condition $y \succeq x$ guarantees that y can be easily transformed into a feasible solution by simply assigning items to slots of larger items. We make the following observation:

Observation 1. *Consider any instance* $1 \ge s_1 \ge \ldots \ge s_n > 0$ *with pattern matrix* A *and vector* $x\in\mathbb{R}^m_{\ge 0}$ *such that* $Ax = \mathbf{1}$. *Then the value of the optimum integral solution to* (4) *and* (1) *coincide.*

However, (4) has the advantage that we can split the solution $x = (x', x'')$ and then separately consider x' and x'' while the vector $b' = Ax'$ might be fractional, which is somewhat unintuitive when speaking about classical bin packing. When $Ax \in \mathbb{Z}^n_{\ge 0}$ and y with $y \succeq x$ is integral, then it is clear that y defines a solution in which each item represented by multiplicity vector Ax can be mapped to one slot in the patterns of y. However, if Ax is fractional, this assignment might be fractional as well. In the following, we discuss the properties of such a mapping.

Consider a vector $x\in\mathbb{R}^m_{\ge 0}$ with a candidate solution $y\in\mathbb{Z}^m_{\ge 0}$. We define a bipartite *supply and demand graph* $G = (U \cup V, E)$ with *demand nodes* $U = \{u_1,\dots,u_n\}$ on the left and *supply nodes* $V = \{v_1,\dots,v_n\}$ on the right. Demand node u_i has a demand of $\mathrm{dem}(i) := Ax_i$ and v_i has a supply of $\sup(i) := A_i y$. Moreover, we have an edge $(u_i, v_j)\in E$ for all $i \ge j$. We should emphasize that it is crucial that the items are sorted according to their sizes.



We state the following very easy lemma (as usual $N(U') := \{v\in V \mid \exists u \in U' : (u,v)\in E\}$ is the set of neighbours of U').

Lemma 4. *Let any instance* $1 \ge s_1 \ge \ldots \ge s_n > 0$ *with vectors* $(x,y)\in\mathbb{R}^m_{\ge 0}\times\mathbb{Z}^m_{\ge 0}$ *be given. Then the following is equivalent:*

a) $x \preceq y$
b) $\sup(\{1,\dots,i\}) \ge \mathit{dem}(\{1,\dots,i\})\ \forall i\in[n]$

c) $\sup(N(U')) \geq dem(U')\ \forall U' \subseteq U$

d) The following transportation problem

$$\begin{aligned} \sum_{j \leq i} z_{ij} &= dem(u_i) \quad \forall i \in [n] \\ \sum_{i \geq j} z_{ij} &\leq \sup(v_j) \quad \forall j \in [n] \\ z_{ij} &\geq 0 \quad \forall (i,j) \in E \end{aligned} \tag{5}$$

has a (fractional) solution z.

Moreover, there is always an optimum integral solution y to (4) *with assignment z satisfying* (5) *such that $z_{ii} \geq \lfloor A_i x \rfloor$ for all $i \in [n]$.*

Proof. The equivalence of *a*) and *b*) is by definition. Hall's theorem provides that *c*) and *d*) are equivalent. To see the equivalence of *b*) and *c*), let $i^* := \max\{i \in U'\}$, then $\text{dem}(\{1,\dots,i^*\}) \geq \text{dem}(U') \geq \sup(N(U')) = \sup(\{v_1 \dots, v_{i^*}\})$.

It remains to show the "moreover" part. Let $y \geq x$ with $y \in \mathbb{Z}^m_{\geq 0}$ be given. We group the fractional parts of $\text{dem}(1),\dots,\text{dem}(n)$ to obtain a new demand function with $\text{dem}'(\{1,\dots,i\}) = \lceil \text{dem}(\{1,\dots,i\}) \rceil$ and $\text{dem}'(i) \in \mathbb{Z}$ for all $i \in [n]$. Since $\sup(j)$ is integral, condition *b*) is still satisfied and Hall's Theorem yields a solution $z \in \mathbb{Z}^E_{\geq 0}$. Now consider the solution y for (4). After replacing patterns with copies, we can assume that $y_p \in \{0,1\}$. For each $j < i$, we replace z_{ij} items of type j by items of type i (note that $s_i \leq s_j$). This results in an integral vector y' that satisfies the claim. □

Notation

To fix some notation, p denotes a pattern which we interpret either as a multi-set of items or as a vector where $p_i \in \mathbb{Z}_{\geq 0}$ denotes the number of copies of item i contained in p. The matrix formed by all possible patterns is denoted by A. Moreover A_i is the ith row of A and by a slight abuse of notation, sometimes we interpret p as a column index and write A^p as the pth column. As usual $[n] = \{1,\dots,n\}$ and $\mathbf{1}$ denotes the all-ones vector of suitable dimension. For a subset $I \subseteq [n]$, we write $s(I) := \sum_{i \in I} s_i$. For any k that is a power of 2, we denote the subset of items $\{i \in [n] \mid \frac{1}{k} \leq s_i < \frac{2}{k}\}$ as one *size class*. The quantity m will usually refer to the number of patterns in A.

5 Operations on fractional solutions

We introduce two useful operations that we can apply to a fractional solution: the classical *item grouping procedure* similar to [FdlVL81, KK82] and a novel *item gluing* operation. Finally, we show how they can be combined to obtain a *well spread* instance in which no pattern contains a significant fraction of copies of a single item.

In order to keep the maintained solution feasible in these procedures it will be necessary to add some additional patterns. In the classical literature [FdlVL81, KK82] this would be done with the phrase "discard the following set of items…" meaning that those items are assigned to separate bins in a greedy manner. We choose to handle this slightly differently. We allow additional columns in A – for each $i \in [n]$, we add a *waste pattern* $\{i\}$, which can be

bought in arbitrary fractional quantities at cost $2s_i$ per copy. For a vector $x \in \mathbb{R}^m_{\geq 0}$ representing a fractional solution, we write

$$(\mathbf{1}, 2s)^T x = \sum_{p \text{ regular pattern}} x_p + \sum_{\{i\} \text{ waste pattern}} 2 s_i x_{\{i\}}$$

as objective function. During our rounding algorithm, we do not make any attempt to round entries $x_{\{i\}}$ belonging to waste patterns to integral values. This can be easily done at the very end as follows:

Lemma 5. *Let $x \in \mathbb{R}^m_{\geq 0}$ and suppose that $|p| = 1$ for all $p \in supp(x)$. Then there is an integral $y \geq x$ with $s^T y \leq s^T x + 1$.*

Proof. By adding dummy copies, we may assume that $x_p = \frac{1}{q}$ for all p (for some large number q). Sort the patterns $p_1, \dots, p_Q$ such that the item sizes in those patterns are non-increasing. Buy each qth pattern starting with p_q plus one copy of p_1. □

Finally, any set of Bin Packing items $S \subseteq [n]$ can be assigned to at most $2\sum_{i \in S} s_i + 1$ bins using a First Fit assignment, which is the reason for the penalty factor of 2 for waste patterns.

5.1 Grouping

The operation of *grouping items* is already defined by de la Vega and Luecker in their asymptotic PTAS for Bin Packing [FdlVL81]. For some parameter k, they form groups of k input items each and round up the item sizes to the size of the largest item in that group. This essentially reduces the number of different item types by a factor of k. In contrast, we will replace items in the *fractional solution* x with *smaller* items. The reason for our different approach is that we measure progress in our algorithm in terms of $|\text{supp}(x)|$, while e.g. Karmarkar-Karp measure the progress in terms of the total size of remaining input items. As a consequence we have to be careful that no operation increases $|\text{supp}(x)|$.

Lemma 6 (Grouping Lemma)**.** *Let $x \in \mathbb{R}^m_{\geq 0}$ be a vector, $\beta > 0$ any parameter and a $S \subseteq [n]$ be a subset of items. Then there is an $x' \geq x$ with identical fractionality as x (except of waste patterns) with $(\mathbf{1}, 2s)^T x' \leq (\mathbf{1}, 2s)^T x + O(\beta \cdot \log(2 \max_{i,j \in S}\{\frac{s_i}{s_j}\})$, and for any $i \in S$, either $A_i x' = 0$ or $s_i A_i x' \geq \beta$.*

Proof. It suffices to consider the case in which $\alpha \leq s_i \leq 2\alpha$ for all $i \in S$ and show that the increase in the objective function is bounded by $O(\beta)$. The general case follows by applying the lemma to all size classes $S \cap \{i \in [n] \mid (\frac{1}{2})^{\ell+1} < s_i \leq (\frac{1}{2})^{\ell}\}$. We also remove those items that have already $s_i A_i x \geq \beta$ from S since there is nothing to do for them.

We consider the index set $I := \{(i,p) \mid i \in S, p \in \text{supp}(x)\}$. For any subset $G \subseteq I$, we define the weight as $w(G) := \sum_{(i,p) \in G} s_i p_i x_p$. Note that any single index has weight $w(\{(i,p)\}) = s_i p_i x_p \leq s_i A_i x \leq \beta$ by assumption. Hence we can partition $I = G_1 \dot\cup \dots \dot\cup G_r$ such that

- $w(G_k) \in [2\beta, 4\beta]\ \forall k = 1, \dots, r-1$
- $w(G_r) \leq 2\beta$
- $(i,p) \in G_k, (i',p') \in G_{k+1} \Rightarrow i \leq i'$

Now, for each $k \in \{1, \dots, r-1\}$ and each index $(i,p) \in G_k$, we replace items of type i in p with the *smallest* item type that appears in G_k. Furthermore, for indices $(i,p) \in G_r$, we remove items of type i from p. Finally, we add $4\frac{\beta}{\alpha}$ many copies of the largest item in I to the waste (note that the number $4\frac{\beta}{\alpha}$ can be fractional and even $4\frac{\beta}{\alpha} \ll 1$ is meaningful). Let x' denote the emerging solution. Clearly, x' only uses patterns that have size at most 1. Moreover, $(\mathbf{1}^T, 2s)x' - (\mathbf{1}^T, 2s)x \le 4\frac{\beta}{\alpha} \cdot 2 \max\{s_i \mid i \in S\} \le 16\beta$.

It remains to argue that $x' \ge x$. Consider any item $i \in [n]$ and the difference $\sum_{j \le i} A_j x' - \sum_{j \le i} A_j x$. There is at most one group G_k whose items were (partly) larger than i in x and then smaller in x'. The weight of that group is $w(G_k) \le 4\beta$, thus their "number" is $\sum_{(i,p)\in G_k} p_i x_p \le \frac{4\beta}{\alpha}$. We add at least this "number" of items to the waste, thus

$$\sum_{j \le i} A_j x' - \sum_{j \le i} A_j x \ge \frac{4\beta}{\alpha} - \sum_{(i,p)\in G_k} p_i x_p \ge 0$$

□

5.2 Gluing

We now formally introduce our novel item gluing method. Assume we would a priori know some set of items which in an optimal integral solution is assigned to the same bin. Then there would be no harm in gluing these items together to make sure they will end up in the same bin. The crucial point is that this is still possible with copies of an item i appearing in the same pattern in a *fractional* solution as long as the contribution $x_p \cdot p_i$ to the multiplicity vector is integral, see again Figure 1.(b).

Lemma 7 (Gluing Lemma)**.** *Suppose that there is a pair of pattern p and item i with $x_p = \frac{r}{q}$ and $p_i \ge w \cdot q$ ($r, q, w \in \mathbb{N}$) as well as a size $s_{i'} = w \cdot s_i$. Modify x such that $w \cdot q$ items of type i in pattern p are replaced by q items of type i' and call the emerging solution x'. Then the following holds:*

a) The patterns in x' have still size at most one and $\mathbf{1}^T x = \mathbf{1}^T x'$.

b) Any integral solution $y' \ge x'$ can be transformed into an integral solution $y \ge x$ of the same cost.

Proof. The first claim is clear as $q \cdot s_{i'} = wq \cdot s_i$.

Now, let $y' \ge x'$ be an integral solution and let z' satisfy the linear system (5) w.r.t. x' and y'. Then $A_{i'} x' \ge \frac{r}{q} \cdot q = r \in \mathbb{Z}_{>0}$. By Lemma 4 we may assume that there are r slots for item i' in patterns in $\mathrm{supp}(y)$. In each of those r pattern p, we resubstitute i' by w copies of i (recall that this is size preserving as $s_{i'} = w \cdot s_i$). □

Observe that since we allow Ax to be fractional, it is crucial that the gluing operation produces integer multiples of an item so that (using the "moreover" part in Lemma 4) we can resubstitute an integer number of items.

Any sequence of grouping and gluing produces a solution which dominates the original instance in the sense that any integral solution for the transformed instance implies an integral solution for the original one.

Corollary 8. *Let $s \in [0,1]^m$ and $x \in \mathbb{R}^m_{\geq 0}$ be any instance for (4). Suppose there is a sequence $x = x^{(0)}, \dots, x^{(T)}$ with $x^{(T)} \in \mathbb{Z}^m_{\geq 0}$ such that for each $t \in \{1, \dots, T\}$, at least one of the cases is true:*

- *(i) $x^{(t)} \succeq x^{(t-1)}$*
- *(ii) $x^{(t)}$ emerges from $x^{(t-1)}$ by gluing items via Lemma 7.*

Then one can construct an integral solution $y \in \mathbb{Z}^m_{\geq 0}$ with $y \geq x$ and $\mathbf{1}^T y \leq \mathbf{1}^T x^{(T)}$ in polynomial time.

Proof. Follows by induction over T, the definition of "$\succeq$" and Lemma 7.b). □

5.3 Obtaining a well-spread instance

As already argued in the introduction, a rounding procedure based on the partial coloring method would beat [KK82] if the patterns in $p \in \text{supp}(x)$ would satisfy $p_i \leq \delta \cdot A_i x$ for $\delta \leq o(1)$, i.e. they are *δ-well-spread.* A crucial lemma is to show that we can combine grouping and gluing to obtain a $\frac{1}{\text{polylog}(n)}$-well-spread solution while loosing a negligible additive $\frac{1}{\text{polylog}(n)}$ term in the objective function.

To simplify notation, let us assume that the vector s contains already all sizes $k \cdot s_i$ for $i = 1, \dots, n$ and $k \in \mathbb{N}$ (even if x does not contain any item of that size).

Lemma 9. *Let $1 \geq s_1 \geq \dots \geq s_n \geq \frac{1}{n}$ and $x \in [0,1]^m$ be given such that for some $q \in \mathbb{Z}_{>0}$ one has $x_p \in \frac{\mathbb{Z}_{\geq 0}}{q}$ for all $p \in supp(x)$ and $|supp(x)| \leq n$. Choose any parameters $\delta, \beta > 0$ and call items of size at least $\varepsilon := \delta \frac{\beta}{2q}$* large *and* small *otherwise. Then one can apply Grouping and Gluing to obtain a solution $\tilde{x}$ with $(\mathbf{1}, 2s)^T \tilde{x} \leq (\mathbf{1}, 2s)^T x + O(\beta \log^2 n)$ and the property that $p_i \leq \delta \cdot A_i \tilde{x}$ for all small items i and all $p \in supp(\tilde{x})$.*

Proof. First apply grouping with parameter β to the small items to obtain a vector $x' \succeq x$ with $(\mathbf{1}, 2s)^T x' \leq (\mathbf{1}, 2s)^T x + O(\beta \cdot \log n)$ such that $s_i \cdot A_i x' \geq \beta$ whenever $A_i x' > 0$. Now apply gluing for each i and $p \in \text{supp}(x')$, wherever $s_i p_i \geq 2\delta \cdot \beta$ with maximal possible w. In fact, that means $w \geq \lfloor \frac{2\delta\beta}{q s_i} \rfloor \geq \frac{\delta\beta}{q s_i}$ since $\frac{\delta\beta}{q s_i} \geq \frac{\delta\beta}{q\varepsilon} \geq 1$. The size of the items emerging from the gluing process is at least $w \cdot s_i \geq \frac{\delta\beta}{q}$, thus they are large by definition. We have at most q items of type i remaining in the pattern and their total size is $q \cdot s_i \leq q \cdot \varepsilon \leq \frac{\delta\beta}{2}$. Let x'' be the new solution.

If after gluing, we still have $s_i A_i x'' \geq \frac{\beta}{2}$, then we say i is *well-covered.* If indeed all small items are well covered, then we are done because $s_i A_{ip} \leq s_i q \leq \frac{\delta\beta}{2} \leq \delta \cdot s_i A_i x''$ for all small i and $p \in \text{supp}(x'')$.

Thus, let $S := \{i \text{ small} \mid i \text{ not well covered}\}$ be the set of those items whose number has decreased to less than half due to gluing. We apply again grouping (Lemma 6) to S (note that we do not touch the well covered items). Then we apply again gluing where ever possible and repeat the procedure until all small items are well covered. Note that once an item is well covered it stays well covered as it is neither affected by grouping nor by gluing.

In each iteration the waste increases by $O(\beta \cdot \log n)$, thus it suffices to argue that the procedure stops after at most $O(\log n)$ iterations. Note that the total size of not well covered items

$\sum_{i \text{ not well covered}} s_i A_i x$ decreases by at least a factor of $\frac{1}{2}$ in each iteration. Moreover at the beginning we had $\sum_{i=1}^n s_i A_i x < n$ and we can stop the procedure when $\sum_{i \text{ not well covered}} s_i A_i x \leq \frac{1}{n^2}$[5], which shows the claim. $\square$

6 The algorithm

In this section, we present the actual rounding algorithm, which can be informally stated as follows (we give a more formal definition later):

(1) FOR $\log n$ iterations DO

 (2) round x s.t. $x_p \in \frac{\mathbb{Z}_{\geq 0}}{\text{polylog}(n)}$ for all p

 (3) make x $\frac{1}{\text{polylog}(n)}$-well spread

 (4) run the constructive partial coloring lemma to make half of the variables integral

For the sake of comparison note that the Karmarkar-Karp algorithm [KK82] consists of step (1) + (4), just that the application of the constructive partial coloring lemma is replaced with grouping + computing a basic solution.

6.1 Finding a partial coloring

The next step is to show how Lemma 1 can be applied to make at least half of the variables integral. As this is the crucial core procedure in our algorithm, we present it as a stand-alone theorem and list all the properties that we need for matrix A. Mathematically speaking, the point is that any matrix A that is column-sparse and has well-spread rows admits good colorings via the entropy method.

Theorem 10. *Let $x \in [0,1]^m$ be a vector and δ, ε be parameters with $0 < \varepsilon \leq \delta^2 \leq 1$ and let $A \in \mathbb{Z}_{\geq 0}^{n \times m}$ (with $m \geq 100 \log(\min_i\{\frac{2}{s_i}\})$) be any matrix with numbers $1 \geq s_1 \geq \ldots \geq s_n > 0$. Suppose that for any column $p \in [m]$, one has $A^p s \leq 1$ and for any row i with $s_i \leq \varepsilon$ one has $\|A_i\|_\infty \leq \delta \cdot \|A_i\|_1$. Then there is a randomized algorithm with expected polynomial running time to compute a $y \in [0,1]^m$ with $|\{j \in [m] \mid y_p \in \{0,1\}\}| \geq \frac{m}{2}$, $\mathbf{1}^T y = \mathbf{1}^T x$ and for all $i \in [n]$*

$$\Big|\sum_{j \leq i} A_j (y - x)\Big| \leq \begin{cases} O(\frac{1}{s_i}) & s_i > \varepsilon \\ O(\sqrt{\ln(\frac{2}{\delta}) \cdot \delta} \cdot \frac{1}{s_i}) & s_i \leq \varepsilon \end{cases}$$

Proof. We will prove the claim via a single application of Lemma 1. In particular we need to make a choice of vectors v_i and parameters λ_i. First of all, it will be convenient for our arguments if each individual row has a small norm. So we can split the rows i for large items (i.e. $s_i > \varepsilon$) so that $\|A_i\|_1 = 1$ and we split the rows for small items so that $\frac{1}{2\delta} \leq \|A_i\|_1 \leq \frac{1}{\delta}$ and $\|A_i\|_\infty = 1$. In the following let $C > 0$ be a large enough constant that we determine later.

First, we partition the items into *groups* $[n] = I_1 \dot\cup \ldots \dot\cup I_t$ such that each group I consists of consecutive items and is chosen maximally such that $\sum_{i \in I} \|A_i\|_1 s_i \leq C$ and so that I contains

[5] In fact, whenever we have a pattern p with $x_p \leq \frac{1}{n}$ we can just move it to the waste. In total over all iterations this does not cost us more than an extra 1 term. Then we always have the trivial lower bound $s_i A_i x \geq \frac{1}{n^2}$ as $s_i \geq \frac{1}{n}$.

only items from one size class. In other words, apart from the $\log(\min_i\{\frac{2}{s_i}\})$ many groups that contain the last items from some size class, we will have that $\sum_{i\in I}\|A_i\|_1 s_i \geq C-1$. For each group I, whether complete or not, we define a vector $v_I := \sum_{i\in I} A_i$ with parameter $\lambda_I := 0$.

Now consider a group I that belongs to small items, say the items $i \in I$ have size $\frac{1}{k} \leq s_i \leq \frac{2}{k}$ for some k. We form *subgroups* $G_1 \subseteq G_2 \subseteq \ldots \subseteq G_{t(I)} = I$ such that $\|v_{G_{j+1}\setminus G_j}\|_1 \in [C\sqrt{\delta}k, 2C\sqrt{\delta}k]$ (this works out since $\|A_i\|_1 \leq \frac{1}{\delta} = \frac{\sqrt{\delta}}{\delta^{3/2}} \leq \frac{\sqrt{\delta}}{\varepsilon} \leq \sqrt{\delta}k$; again the last subgroup $G_{t(I)}$ might be smaller). For each subgroup G, we add the vector v_G to our list, equipped with error parameter $\lambda_G := 4\sqrt{\ln(\frac{2}{\delta})}$.

To control the objective function, we also add the all-ones vector $v_{\text{obj}} := (1,\ldots,1)$ with $\lambda_{\text{obj}} := 0$. Now we want to argue that Lemma 1, applied to all the vectors v_I, v_G, v_{obj} defined above, provides a solution y that satisfies the claim. The first step is to verify that indeed the "entropy condition" (3) is satisfied. As the item sizes for each column sum up to most one, we know that $\sum_{i\in[n]}\|A_i\|_1 s_i \leq m$, where m is the number of columns. Each complete group has $\sum_{i\in I}\|A_i\|_1 s_i \geq C-1$, thus the number of groups is $t \leq \frac{m}{C-1} + \log(\min_i\{\frac{2}{s_i}\}) \leq \frac{m}{50}$ for C large enough and each group I contributes $e^{-\lambda_I^2/16} = 1$ to (3).

Next, consider any group I and let us calculate the contribution just of its subgroups $G_1,\ldots,G_{t(I)}$ to (3). The number of I's subgroups is $t(I) \leq \frac{1}{\sqrt{\delta}} + 1 \leq \frac{2}{\sqrt{\delta}}$ and each subgroup G contributes $e^{-\lambda_G^2/16} = \frac{\delta}{2}$, thus their total contribution is bounded by $\frac{2}{\sqrt{\delta}}\cdot\frac{\delta}{2} \leq 1$. In other words, the total contribution of all subgroups is bounded from above by $\frac{m}{100}$ as well and (3) indeed holds and we can apply Lemma 1. The algorithm returns a vector y such that $|v_I(x-y)| = 0$ for each group I and $|v_G(x-y)| \leq 4\sqrt{\ln(\frac{2}{\delta})}\cdot\|v_G\|_2$ for each subgroup G (and of course $\mathbf{1}^T x = \mathbf{1}^T y$).

Finally, consider any item i and suppose it is small with $\frac{1}{k} \leq s_i \leq \frac{2}{k}$. It remains to show that $|\sum_{j\leq i} A_j(x-y)| \leq O(\sqrt{\delta\ln(\frac{2}{\delta})}\cdot k)$. Now we use that the interval $\{1,\ldots,i\}$ can be written as disjoint union of a couple of groups + a single subgroup + a small rest. So let t' be the index with $i \in I_{t'}$. Moreover, let G be the (unique) maximal subgroup such that $G \subseteq \{1,\ldots,i\}\setminus\bigcup_{t''<t'} I_{t''}$ and let $R := \{1,\ldots,i\}\setminus(G\cup\bigcup_{t''<t'} I_{t''})$ be the remaining row indices. The error that our rounding produces w.r.t. i is

$$\Big|\sum_{j\leq i} A_j(x-y)\Big| = \Big|\sum_{t''<t'} \underbrace{v_{I_{t''}}(x-y)}_{=0} + v_G(x-y) + v_R(x-y)\Big| \leq 4\sqrt{\ln\left(\frac{2}{\delta}\right)}\cdot\|v_G\|_2 + \underbrace{\|v_R\|_1}_{\leq 2C\sqrt{\delta}k}$$

It remains to bound $\|v_G\|_2$. At this point, we crucially rely on the assumption $\|A_i\|_\infty \leq 2\delta\cdot\|A_i\|_1$. Using this together with Hölder's inequality and the triangle inequality we obtain

$$\|v_G\|_2 \leq \sqrt{\|v_G\|_1\cdot\|v_G\|_\infty} \leq \sqrt{\|v_G\|_1\cdot\sum_{i\in G}\|A_i\|_\infty} \leq \sqrt{\|v_G\|_1\cdot 2\delta\sum_{i\in G}\|A_i\|_1} = \sqrt{2\delta}\underbrace{\|v_G\|_1}_{\leq O(k)} = O(\sqrt{\delta}k)$$

and the claim is proven. Note that for large items i we can satisfy $|\sum_{j\leq i} A_j(x-y)| \leq O(\frac{1}{s_i})$ even without using subgroups. $\square$

One can alternatively prove Theorem 10 by combining the classical partial coloring lemma and the Lovász-Spencer-Vesztergombi Theorem [LSV86]. Note that the bound of the classical partial coloring lemma involves an extra $O(\log\frac{1}{\lambda})$ term for $\lambda \leq 2$. However, using the

parametrization as in [SST] one could avoid loosing a super constant factor. Moreover, instead of just 2 different group types ("groups" and "subgroups"), we could use an unbounded number to save the $\sqrt{\ln(\frac{2}{\delta})}$ factor. However, this would not improve on the overall approximation ratio.

Recall that for the gluing procedure in Lemma 9 we need the property that the entries in x_p are not too tiny – say at least $\frac{1}{\text{polylog}(n)}$. But this is easy to achieve (in fact, the next lemma also follows from [KK82] or [Rot12]).

Lemma 11. *Given any instance* $1 \geq s_1 \geq \ldots \geq s_n \geq \frac{1}{n}$, $x \in \mathbb{R}^m_{\geq 0}$ *and parameter* $\gamma > 0$. *Then one can compute a* $y \geq x$ *in expected polynomial time such that all* y_p *are multiples of* γ *and* $(\mathbf{1}, 2s)^T y \leq (\mathbf{1}, 2s)^T x + O(\gamma \cdot \log^2 n)$.

Proof. After replacing x with a basic solution, we may assume that $|\text{supp}(x)| \leq n$. We write $x = \gamma x' + \gamma z$ with $z \in \mathbb{Z}^m_{\geq 0}$ and $\mathbf{0} \leq x' \leq \mathbf{1}$. Now apply $\log n$ times Theorem 10 with $\delta = 1$ to x' to obtain $y' \in \{0,1\}^m$ with $\mathbf{1}^T y' \leq \mathbf{1}^T x'$ and $|\sum_{j \leq i} A_j(x - y)| \leq O(\log n \cdot \frac{1}{s_i})$ (if at the end of the rounding process, the fractional support goes below $100 \log(n)$, we can stop and remove the remaining fractional patterns). Let i_ℓ be the largest item in size class ℓ. Then for $\ell = 0, \ldots, \log n$, we add $O(\log n \cdot 2^\ell)$ copies of item i_ℓ to the waste of y'. Now $(\mathbf{1}, 2s)^T y' \leq (\mathbf{1}, 2s)^T x' + O(\log^2 n)$ and $y' \geq x'$. Eventually define $y := \gamma y' + \gamma z$ and observe that $y \geq x$, all entries y_p are multiples of γ and $(\mathbf{1}, 2s)^T y \leq (\mathbf{1}, 2s)^T x + O(\gamma \log^2 n)$. $\square$

Observe that just applying Lemma 11 with $\gamma = 1$ yields an integral solution with cost $OPT_f + O(\log^2 n)$.

6.2 Proof of the main theorem

It remains to put all ingredients together and show that each of the $\log n$ applications of the partial coloring lemma increases the objective function by at most $O(\log \log n)$.

Theorem 12. *Let* $1 \geq s_1 \geq \ldots \geq s_n \geq \frac{1}{n}$ *with and* $x \in \mathbb{R}^m_{\geq 0}$ *be given. Then there is an expected polynomial time algorithm to compute* $y \geq x$ *with* $\mathbf{1}^T y \leq \mathbf{1}^T x + O(\log n \cdot \log \log n)$.

Proof. We choose $\delta := \beta := \gamma := \frac{1}{\lceil \log^4 n \rceil}$ and $\varepsilon := \frac{1}{4 \log^{12} n} \leq \frac{1}{2} \gamma \beta \delta$. After moving to a basic solution and buying integral parts of x, we may assume that $x \in [0,1]^m$ and $|\text{supp}(x)| \leq n$. Recall that $i_\ell = \text{argmax}\{i \mid 2^{-(\ell+1)} < s_i \leq 2^{-\ell}\}$. We perform the following algorithm:

(1) FOR $t = 1$ TO $\log n$ DO

  (2) Apply Lemma 11 to have all x_p being multiples of γ.

  (3) Apply Lemma 9 to make x δ-well spread for items of size at most ε.

  (4) Apply Theorem 10 to x to halve the number of fractional entries[6] (if $|\text{supp}(x)| \leq 100 \log(n)$, just set $x_p := 1$ for all patterns in the support).

  (5) FOR each size class ℓ, add $O(2^\ell)$ items of i_ℓ to the waste if $s_{i_\ell} \geq \varepsilon$ and $O(\sqrt{\delta \ln(\frac{2}{\delta})} \cdot 2^\ell)$ items if $s_{i_\ell} < \varepsilon$.

[6] To be precise, we apply Theorem 10 for the submatrix of A corresponding to the columns in $\text{supp}(x)$.

(6) Resubstitute glued items in x and replace waste patterns via a greedy assignment to obtain y.

Let $x^{(t)}$ be the value of x at the end of the tth while loop. Let $x^{(t,2)}$ be the solution x at the *end* of step (2) in the tth iteration. We define $x^{(t,3)}, x^{(t,4)}, x^{(t,5)}$ analogously. Furthermore, $x^{(t,1)}$ be x at the beginning of the while loop.

First of all, observe that $x^{(t,2)}$ satisfies $x^{(t,2)} \geq x^{(t,1)}$ and $x_p^{(t,2)} \in \gamma\mathbb{Z}_{\geq 0}$ for all p, by the properties of Lemma 11. The vector $x^{(t,3)}$ emerges from $x^{(t,2)}$ by grouping and gluing and according to Lemma 9, it satisfies $A_{ip} \leq \delta \cdot A_i x^{(t,3)} \leq \delta \cdot \sum_{\tilde{p} \in \mathrm{supp}(x^{(t,3)})} \tilde{p}_i$ for all $p \in \mathrm{supp}(x^{(t,3)})$ and all i with $s_i \leq \varepsilon \leq \frac{1}{2}\gamma\beta\delta$. Finally, the conditions of Theorem 10 are satisfied by parameters δ and ε, thus the extra items bought in step (5) are enough to have $x^{(t,5)} \geq x^{(t,3)}$. Note that none of the steps (2),(3),(5) increases the number of regular patterns in the support, but $|\{p \mid x_p^{(t,4)} \notin \{0,1\}\}| \leq \frac{1}{2}|\{p \mid x_p^{(t,3)} \notin \{0,1\}\}|$, thus $x^{(\log n)}$ is indeed integral.

Hence, by Corollary 8 we know that y will be a feasible solution to the original bin packing instance of cost at most $(\mathbf{1}, 2s)^T x^{(\log n)} + 1$. It remains to account for the increase in the objective function. Each application of (2) increases the objective function by at most $O(\gamma \log^2 n)$. (3) costs us $O(\beta \log^2 n)$ and (4) + (5) increases the objective function by $O(\log \frac{1}{\varepsilon} + \sqrt{\delta \ln(\frac{2}{\delta})} \log n)$. In total over $\log n$ iterations,

$$\mathbf{1}^T y - \mathbf{1}^T x \leq O(\gamma \log^3 n) + O(\beta \log^3 n) + O(\log n \cdot \log \tfrac{1}{\varepsilon}) + O(\sqrt{\delta \ln(\tfrac{2}{\delta})} \log^2 n) \leq O(\log n \cdot \log\log n)$$

plugging in the choices for $\delta, \beta, \gamma, \varepsilon$. $\square$

Together with the remark from Lemma 3, the approximation guarantee for our main result, Theorem 2 follows. Let us conclude with a quick estimate on the running time. Given a Bin Packing instance $s_1, \ldots, s_n$ (with one copy of each item, so n is the total number of items), one can compute a fractional solution x of cost $\mathbf{1}^T x \leq (1+\varepsilon) OPT_f$ in time $O((\frac{n^2}{\varepsilon^4} + \frac{1}{\varepsilon^6}) \log^5(\frac{n}{\varepsilon}))$ with $|\mathrm{supp}(x)| \leq n$ (see Theorem 5.11 in [PST95]). We set $\varepsilon := \frac{1}{n}$ and obtain an x with $\mathbf{1}^T x \leq OPT_f + 1$ in time $O(n^6 \log^5(n))$. It suffices to run the Constructive Partial Coloring Lemma with error parameter $\delta := \frac{1}{n}$, which takes time $O(\frac{\tilde{n}^3}{\delta^2} \log \frac{\tilde{n}\tilde{m}}{\delta}) = \tilde{O}(n^5)$ where $\tilde{n} \leq n \cdot \mathrm{polylog}(n)$ is the number of vectors and $\tilde{m} \leq n$ is the dimension of x. In other words, the running time is dominated by the computation of the fractional solution. Finally we obtain a vector y with $y_p \in [0, \frac{1}{n}] \cup [1 - \frac{1}{n}, 1]$ and $|\mathrm{supp}(y)| \leq n$. We move those entries with $y_p \leq \frac{1}{n}$ to the waste, increasing the objective function by at most $n \cdot \frac{1}{n} = 1$ and we roundup those entries with $y_p \geq 1 - \frac{1}{n}$.

7 Remarks

The obvious question is, how tight is our analysis? Well, suppose that our instance has only items of size at least $\frac{1}{(\log n)^{1/10}}$, then it is not clear whether our gluing approach might be applicable or not.

Another interesting observation concerning the application of the Constructive Partial Coloring Lemma is the following: recall that we gave vectors v_I for groups and vectors v_G for all subgroups as input to Lemma 1. But the solution y returned by that lemma is the end point of a Brownian motion and satisfies $\Pr[|v(x-y)| \leq \lambda \|v\|_2] \leq e^{-\Omega(\lambda^2)}$ for every $\lambda \geq 0$

and $v \in \mathbb{R}^m$ regardless whether v is known to the algorithm or not. If we choose $\lambda_G := \log n$ (and $\delta, \gamma, \varepsilon$ slightly more generous), then the guarantee $|v_G(x-y)| \leq \lambda_G \|v_G\|$ is satisfied for all subgroups with high probability anyway and there is no need to include them in the input.

Moreover, we are not even using the full power of the constructive partial coloring lemma. Suppose we had only a weaker Lemma 1 which needs the stronger assumption that for example $\sum_i (1+\lambda_i)^{-10} \leq \frac{m}{16}$ instead of the exponential decay in λ. We would still obtain the same asymptotic bound of $O(\log n \cdot \log\log n)$, thus a fine tuning of parameters is not going to give any improvement.

Consider the seemingly simple *3-Partition* case in which all n items have size $\frac{1}{4} < s_i < \frac{1}{2}$. Both, the approaches of Karmarkar and Karp [KK82] and ours provide a $O(\log n)$ upper bound on the integrality gap. The construction of Newman and Nikolov [NNN12] of 3 badly colorable permutations can be used to define a 3-Partition instance with an optimum fractional solution $x \in \{0, \frac{1}{2}\}^m$ such that any integral solution $y \in \mathbb{Z}^m_{\geq 0}$ with $\mathrm{supp}(y) \subseteq \mathrm{supp}(x)$ and $\mathbf{1}^T y - \mathbf{1}^T x \leq o(\log n)$ satisfies $\max_{i \in [n]} \{\sum_{j \leq i} (A_j x - A_j y)\} \geq \Omega(\log n)$. This suggests that either the $\log n$ bound is best possible for 3-Partition or some fundamentally new ideas are needed to make progress.

Acknowledgements. The author is grateful to Michel X. Goemans for helpful discussions and support and to Nikhil Bansal for reading a preliminary version.

References

[AHK12] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

[Ban98] W. Banaszczyk. Balancing vectors and Gaussian measures of n-dimensional convex bodies. *Random Structures Algorithms*, 12(4):351–360, 1998.

[Ban10] N. Bansal. Constructive algorithms for discrepancy minimization. In *FOCS*, pages 3–10, 2010.

[Bec81] J. Beck. Roth's estimate of the discrepancy of integer sequences is nearly sharp. *Combinatorica*, 1(4):319–325, 1981.

[Bec88] J. Beck. Irregularities of distribution ii. *Proc. London Math. Soc. (3), 56:1?50*, 1988.

[BF81] J. Beck and T. Fiala. "Integer-making" theorems. *Discrete Appl. Math.*, 3(1):1–8, 1981.

[CGJ84] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin-packing—an updated survey. In *Algorithm design for computer system design*, volume 284 of *CISM Courses and Lectures*, pages 49–106. Springer, Vienna, 1984.

[Dós07] G. Dósa. The tight bound of first fit decreasing bin-packing algorithm is FFD(I) <= 11/9OPT(I) + 6/9. In B. Chen, M. Paterson, and G. Zhang, editors, *ESCAPE*, volume 4614 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2007.

[Eis57] K. Eisemann. The trim problem. *Management Science*, 3(3):279–284, 1957.

[EPR11] F. Eisenbrand, D. Pálvölgyi, and T. Rothvoß. Bin packing via discrepancy of permutations. In *SODA*, pages 476–481, 2011.

[EPR13] F. Eisenbrand, D. Pálvölgyi, and T. Rothvoß. Bin packing via discrepancy of permutations. In *Accepted to Transactions on Algorithms (Special Issue for SODA 2011)*, 2013.

[FdlVL81] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

[GG61] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9:849–859, 1961.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, New York, 1979.

[GLS81] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[JDU+74] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3(4):299–325, 1974.

[Joh73] D. S. Johnson. *Near-optimal bin packing algorithms*. PhD thesis, MIT, Cambridge, MA, 1973.

[JSO10] K. Jansen and R. Solis-Oba. An OPT + 1 algorithm for the cutting stock problem with constant number of object lengths. In Friedrich Eisenbrand and F. Bruce Shepherd, editors, *Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization, IPCO 2010*, pages 438–449, 2010.

[KK82] N. Karmarkar and R. M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *23rd annual symposium on foundations of computer science (Chicago, Ill., 1982)*, pages 312–320. IEEE, New York, 1982.

[LM12] S. Lovett and R. Meka. Constructive discrepancy minimization by walking on the edges. In *FOCS*, pages 61–67, 2012.

[LSV86] L. Lovász, J. Spencer, and K. Vesztergombi. Discrepancy of set-systems and matrices. *European J. Combin.*, 7(2):151–160, 1986.

[Mat99] J. Matoušek. *Geometric discrepancy*, volume 18 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1999. An illustrated guide.

[NNN12] A. Newman, O. Neiman, and A. Nikolov. Beck's three permutations conjecture: A counterexample and some consequences. In *FOCS*, pages 253–262, 2012.

[PST95] S. A. Plotkin, D. B. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math. Oper. Res.*, 20(2):257–301, 1995.

[Rot12] T. Rothvoß. The entropy rounding method in approximation algorithms. In *SODA*, pages 356–372, 2012.

[Spe85] J. Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289(2):679–706, 1985.

[Sri97] A. Srinivasan. Improving the discrepancy bound for sparse matrices: Better approximations for sparse lattice approximation problems. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'97 (New Orleans, Louisiana, January 5-7, 1997)*, pages 692–701, Philadelphia, PA, 1997. ACM SIGACT, SIAM, Society for Industrial and Applied Mathematics.

[SS09] A. Sebő and G. Shmonin. Proof of the modified integer round-up conjecture for bin packing in dimension 7. Personal communication, 2009.

[SST] J. H. Spencer, A. Srinivasan, and P. Tetali. The discrepancy of permutation families. Unpublished manuscript.

[ST97] G. Scheithauer and J. Terno. Theoretical investigations on the modified integer round-up property for the one-dimensional cutting stock problem. *Operations Research Letters*, 20(2):93 – 100, 1997.

[WS11] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, 2011.